

**What's on this page:** Every command absent from the page 1 sheet — including newly shipped commands, platform-specific commands, prompt-engineering "cheat codes" the community discovered, useful env vars, and recently deprecated/changed commands. Badges: **NEW** = added in recent release **PRO/MAX** = plan-gated **PLATFORM** = macOS/Win only **SKILL** = bundled skill **CHANGED** = deprecated or behavior changed.

### OMITTED COMMANDS — A TO M

<code>/autofix-pr [prompt]</code>	Spawn a web session that auto-fixes CI failures & PR review comments. Requires <code>gh</code> CLI. <b>NEW</b>
<code>/bug [report]</code>	Submit feedback or a bug report to Anthropic / feedback
<code>/desktop</code>	Continue this session in the Claude Code Desktop app / app <b>MACOS/WIN</b>
<code>/extra-usage</code>	Configure pay-as-you-go overflow when plan rate limit is hit
<code>/feedback [report]</code>	Submit feedback to Anthropic /bug
<code>/mobile</code>	Show QR code to download the Claude mobile app / ios /android
<code>/passes</code>	Share a free week of Claude Code with friends (eligible accounts only) <b>PRO/MAX</b>

### OMITTED COMMANDS — P TO Z

<code>/powerup</code>	Interactive feature lessons with animated demos — learn Claude Code fast <b>NEW</b>
<code>/privacy-settings</code>	View & update data privacy prefs <b>PRO/MAX</b>
<code>/remote-env</code>	Configure the default remote environment for web sessions started with <code>--remote</code> <b>NEW</b>
<code>/setup-bedrock</code>	Wizard to configure Amazon Bedrock auth, region, model pins
<code>/setup-vertex</code>	Wizard to configure Google Vertex AI auth & model pins
<code>/stats</code>	Open usage dashboard (alias for /usage cost tab) / usage <b>NEW V2.1.118</b>
<code>/themes</code>	Open theme picker; manage custom themes via <code>~/.claude/themes/</code> <b>NEW V2.1.118</b>
<code>/upgrade</code>	Upgrade Claude Code to the latest version <b>PRO/MAX</b>

### AUTO MODE & AUTONOMY

SESSION FLAGS (RESEARCH PREVIEW)	
<code>/plan [desc]</code>	Read-only reasoning — plans before any changes. <code>Shift+Tab</code> cycles into this mode
<code>/autofix-pr</code>	Autonomous PR watch — fixes CI failures & review comments continuously <b>NEW</b>
<code>/loop</code>	Omit prompt → runs <code>~/.claude/loop.md</code> or autonomous maintenance check /proactive
TASK MANAGEMENT (V2.1.16+)	
<code>Ctrl+T</code>	Toggle persistent task list — survives session close
"Add [task]"	Natural language → Claude decomposes into a tracked task list
SUB-AGENT CLI	
<code>--agents '{...}'</code>	Define custom sub-agent roles at launch via JSON

### CHANGED & DEPRECATED

<code>/review</code>	<b>DEPRECATED</b> Removed. Install the <code>code-review</code> plugin instead: <code>claude plugin install code-review@claude-plugins-official</code>
<code>/pr-comments</code>	<b>REMOVED V2.1.91</b> Ask Claude directly to view PR comments instead
<code>/rewind</code>	<b>ENHANCED</b> Now rewinds conversation and/or code <i>separately</i> — choose at the prompt
<code>/simplify</code>	<b>SKILL</b> Replaced /review — runs 3-agent parallel code review
<code>/usage</code>	<b>V2.1.118</b> Now a unified dashboard — cost, rate limits, daily stats. <code>/cost</code> & <code>/stats</code> open specific tabs
<code>.claude/commands/</code>	<b>MERGED</b> Custom commands merged into skills. Files still work but <code>~/.claude/skills/</code> is now preferred

### OUTPUT STYLES & THEMES

<code>/output-style</code>	Change Claude's response format and verbosity style
<code>/themes</code>	Open the theme picker UI <b>V2.1.118</b>
OUTPUT STYLE PRESETS	
<b>Concise</b>	Minimal prose, results only
<b>Educational</b>	Explains reasoning at each step
<b>Code Reviewer</b>	Structured critique format
<b>Custom</b>	Define your own in <code>~/.claude/output-modes/</code>
THEME CUSTOMIZATION	
<code>~/.claude/themes/</code>	JSON files define custom themes picked up automatically

### COMMUNITY PROMPT CODES prepend to any prompt

REASONING DEPTH	
<b>L99</b>	Max reasoning depth — expert-level analysis for hard problems
<code>/deepthink</code>	Force a longer reasoning chain before answering
<b>Think step by step</b>	Classic CoT trigger — reliable for multi-stage reasoning
PROBLEM-SOLVING FRAMEWORKS	
<b>OODA</b>	Observe → Orient → Decide → Act — great for decisions
<b>INVERT</b>	Solve backwards from the goal — surfaces hidden assumptions
<b>XRAY</b>	Look past the obvious answer to find the real problem
<b>SCAFFOLD</b>	Project setup mode — structure first, then fill in details
OUTPUT SHAPING	
<b>PERSONA</b>	Answer as a named domain expert (e.g. <code>PERSONA: senior_SRE</code> )
<code>/ghost</code>	Rewrite output in natural human tone (removes AI cadence)

## KEY ENVIRONMENT VARIABLES

set before launching claude

### CONTEXT & MEMORY

<b>CLAUDE_CODE_NEW_INIT=1</b>	Interactive /init flow covering skills, hooks & personal memory
<b>CLAUDE_CODE_ADDITIONAL_DIRECTORIES_CLAUDE_MD=1</b>	Load CLAUDE.md files from --add-dir directories (off by default)
<b>SLASH_COMMAND_TOOL_CHAR_BUDGET=N</b>	Override the 8,000-char skill description budget (default 1% of context)

### CLOUD & REMOTE

<b>CLAUDE_CODE_USE_BEDROCK=1</b>	Route requests through Amazon Bedrock (enables /setup-bedrock wizard)
<b>CLAUDE_CODE_USE_VERTEX=1</b>	Route requests through Google Vertex AI (enables /setup-vertex wizard)

### BEHAVIOR & OUTPUT

<b>ANTHROPIC_MODEL</b>	Default model to use (overrides built-in default)
<b>CLAUDE_CODE_MAX_OUTPUT_TOKENS</b>	Cap max output tokens per response
<b>DISABLE_AUTOUPDATER=1</b>	Disable automatic Claude Code self-updates
<b>DISABLE_NON_ESSENTIAL_TRAFFIC=1</b>	Block telemetry & non-essential outbound calls (air-gap friendly)

### DEBUGGING

<b>cclaude --debug</b>	Enable debug logging from session start (vs. /debug which starts mid-session)
------------------------	---

## /COPY POWER FEATURES

<b>/copy</b>	Copy last response to clipboard
<b>/copy 2</b>	Copy the 2nd-to-last response
<b>/copy → w</b>	In the code-block picker, press <b>w</b> to write directly to a file (great over SSH)

### /CONTEXT DETAILS

<b>/context</b>	Shows optimization suggestions: context-heavy tools, memory bloat, capacity warnings — not just a usage bar
-----------------	---

### /DOCTOR DETAILS

<b>/doctor → f</b>	Press <b>f</b> after running /doctor to have Claude auto-fix reported issues
--------------------	--

## /LOOP & /SCHEDULE DEEP DIVE

<b>/loop 5m [prompt]</b>	Runs prompt every 5 min while session is open
<b>/loop [prompt]</b>	No interval → Claude self-paces between iterations
<b>/loop</b>	No prompt → runs <code>.claude/loop.md</code> or autonomous maintenance check
<b>/proactive</b>	Alias for /loop /loop
<b>/schedule [desc]</b>	Cloud-persisted routines — survives session close. Claude walks you through setup

### CRON EQUIVALENT (CI/SCRIPTING)

```
at / cron + -p echo 'cd /proj && claude -p "$(cat p.md)"' | at 21:30
```

## /BATCH & /AUTOFIX-PR

<b>/BATCH — PARALLEL CODEBASE CHANGES</b>	
<b>/batch &lt;instruction&gt;</b>	Decomposes work into 5–30 units → spawns one agent per unit in isolated git worktrees → each opens its own PR
<b>Example</b>	<code>/batch migrate src/ from Solid to React</code>
<b>/AUTOFIX-PR — CI WATCH</b>	
<b>/autofix-pr</b>	Watches current branch PR. Pushes fixes when CI fails or reviewers comment. Detects PR from checked-out branch
<b>/autofix-pr [prompt]</b>	Pass custom instructions, e.g. <code>/autofix-pr only fix lint errors</code>
<b>Requirement</b>	<code>gh</code> CLI must be installed & authenticated

### /powerup is the fastest onboarding

Runs animated interactive lessons on features you haven't used. Better than reading release notes.

### L99 vs /deepthink

Both push harder reasoning. L99 is a community convention; /deepthink is a recognized prompt modifier. Use L99 for architecture, /deepthink for step-by-step debugging.

### /loop without a prompt

Drops into autonomous maintenance mode — Claude decides what to check. Great for long-running monitoring tasks.

### Prompt codes ≠ magic

OODA, INVERT, L99 work by giving Claude a clearer reasoning framework — not by unlocking hidden modes. They're prompt engineering, not backdoors.